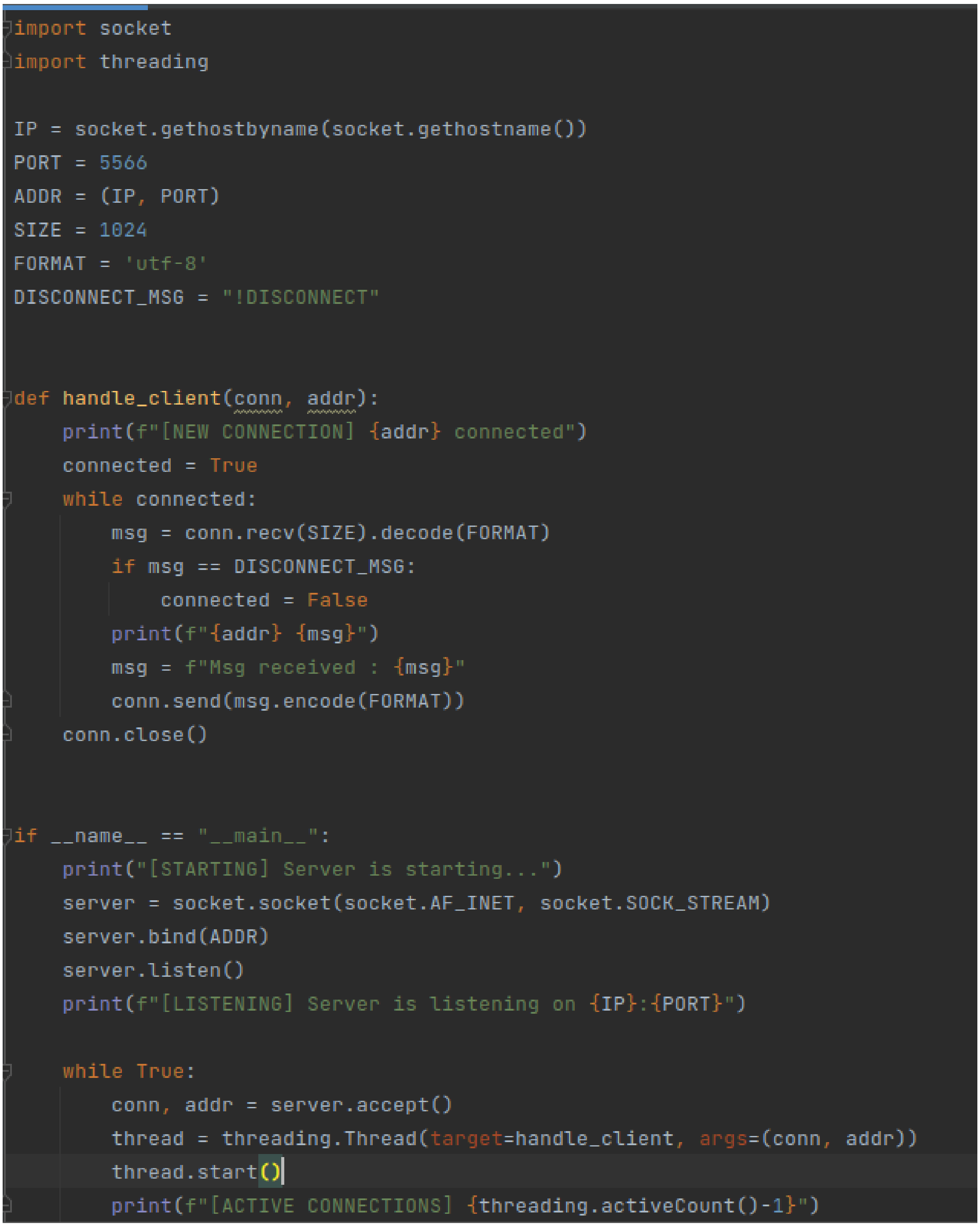
Assignment-3

Name: S.Vishnu Teja **Roll: CS21B2037**

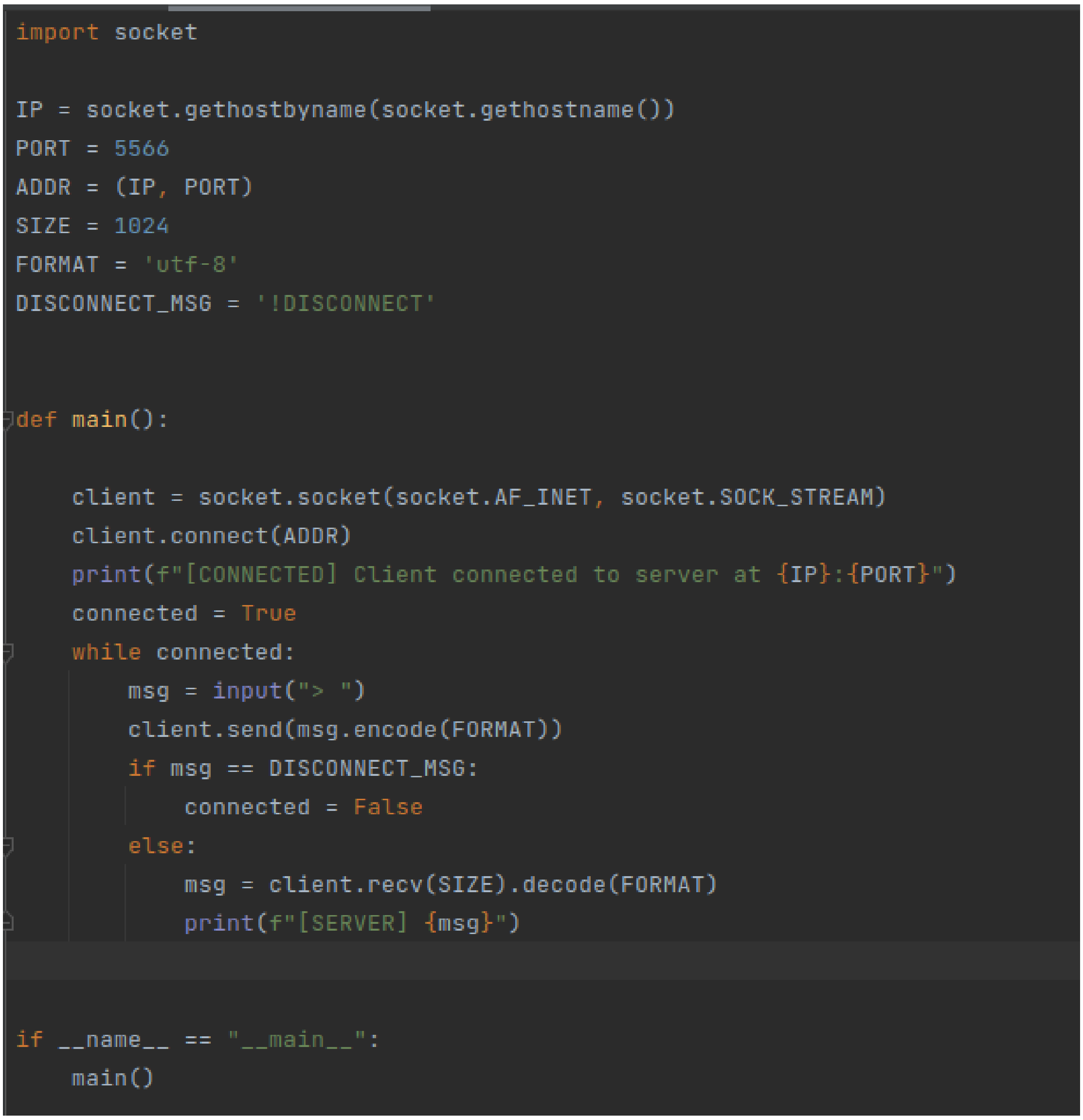
1. Demonstrate with Local Loop

Server Code:-



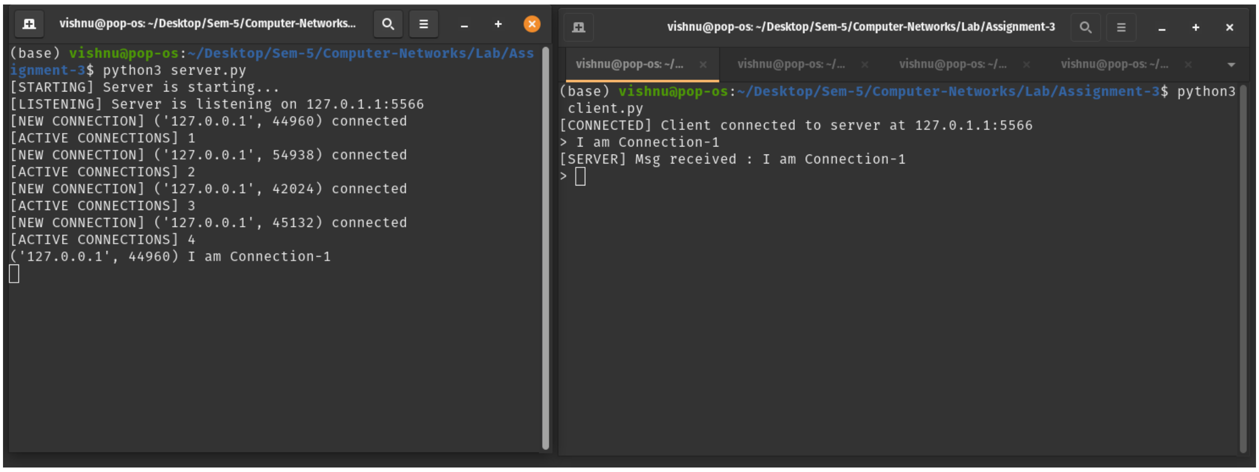
```
import socket
import threading

IP = socket.gethostbyname(socket.gethostname())
PORT = 5566
ADDR = (IP, PORT)
SIZE = 1024
FORMAT = 'utf-8'
DISCONNECT_MSG = "!DISCONNECT"


def handle_client(conn, addr):
    print(f"[NEW CONNECTION] {addr} connected")
    connected = True
    while connected:
        msg = conn.recv(SIZE).decode(FORMAT)
        if msg == DISCONNECT_MSG:
            connected = False
        print(f"{addr} {msg}")
        msg = f"Msg received : {msg}"
        conn.send(msg.encode(FORMAT))
    conn.close()


if __name__ == "__main__":
    print("[STARTING] Server is starting...")
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(ADDR)
    server.listen()
    print(f"[LISTENING] Server is listening on {IP}:{PORT}")

    while True:
        conn, addr = server.accept()
        thread = threading.Thread(target=handle_client, args=(conn, addr))
        thread.start()
        print(f"[ACTIVE CONNECTIONS] {threading.activeCount()-1}")
```

Client Code:-



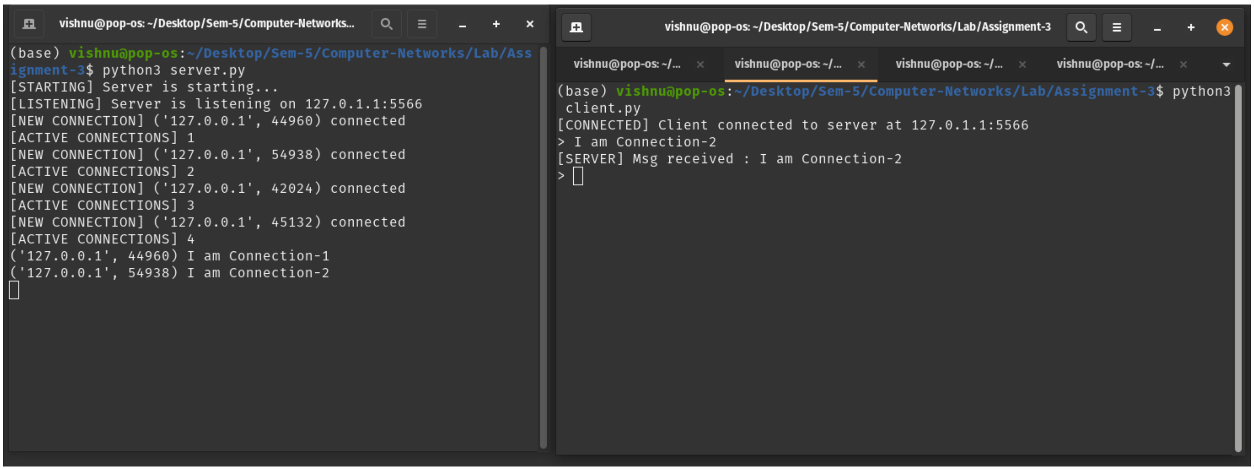
```python
import socket

IP = socket.gethostbyname(socket.gethostname())
PORT = 5566
ADDR = (IP, PORT)
SIZE = 1024
FORMAT = 'utf-8'
DISCONNECT_MSG = '!DISCONNECT'


def main():

    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(ADDR)
    print(f"[CONNECTED] Client connected to server at {IP}:{PORT}")
    connected = True
    while connected:
        msg = input("> ")
        client.send(msg.encode(FORMAT))
        if msg == DISCONNECT_MSG:
            connected = False
        else:
            msg = client.recv(SIZE).decode(FORMAT)
            print(f"[SERVER] {msg}")


if __name__ == "__main__":
    main()
```

Outputs:-


vishnu@pop-os: ~/Desktop/Sem-5/Computer-Networks...
vishnu@pop-os: ~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...

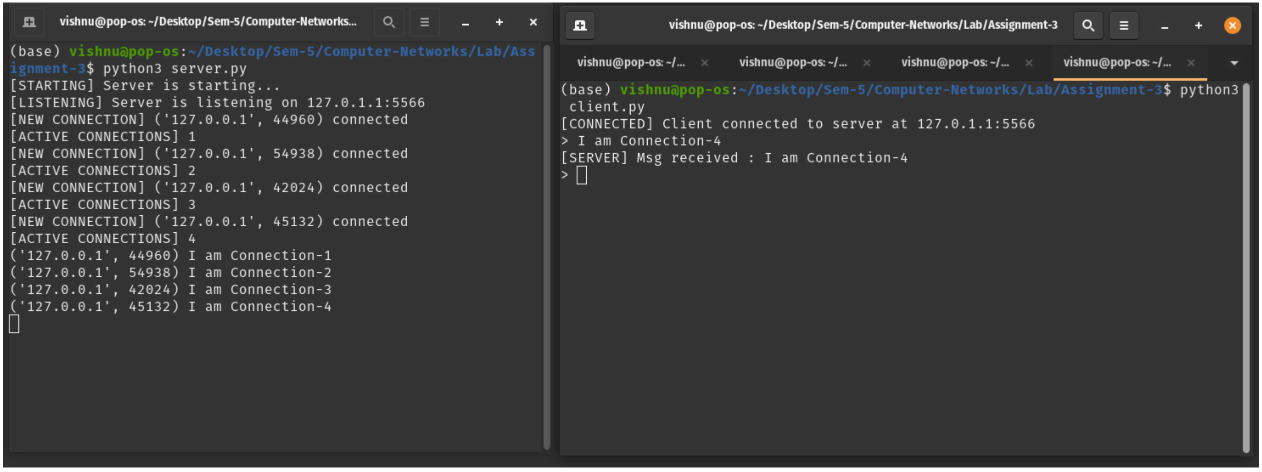
```
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Ass
ignment-3$ python3 server.py
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.1.1:5566
[NEW CONNECTION] ('127.0.0.1', 44960) connected
[ACTIVE CONNECTIONS] 1
[NEW CONNECTION] ('127.0.0.1', 54938) connected
[ACTIVE CONNECTIONS] 2
[NEW CONNECTION] ('127.0.0.1', 42024) connected
[ACTIVE CONNECTIONS] 3
[NEW CONNECTION] ('127.0.0.1', 45132) connected
[ACTIVE CONNECTIONS] 4
('127.0.0.1', 44960) I am Connection-1
```

```
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3$ python3
 client.py
[CONNECTED] Client connected to server at 127.0.1.1:5566
> I am Connection-1
[SERVER] Msg received : I am Connection-1
> 
```


vishnu@pop-os: ~/Desktop/Sem-5/Computer-Networks...
vishnu@pop-os: ~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...

```
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Ass
ignment-3$ python3 server.py
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.1.1:5566
[NEW CONNECTION] ('127.0.0.1', 44960) connected
[ACTIVE CONNECTIONS] 1
[NEW CONNECTION] ('127.0.0.1', 54938) connected
[ACTIVE CONNECTIONS] 2
[NEW CONNECTION] ('127.0.0.1', 42024) connected
[ACTIVE CONNECTIONS] 3
[NEW CONNECTION] ('127.0.0.1', 45132) connected
[ACTIVE CONNECTIONS] 4
('127.0.0.1', 44960) I am Connection-1
('127.0.0.1', 54938) I am Connection-2
```

```
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3$ python3
 client.py
[CONNECTED] Client connected to server at 127.0.1.1:5566
> I am Connection-2
[SERVER] Msg received : I am Connection-2
>
```

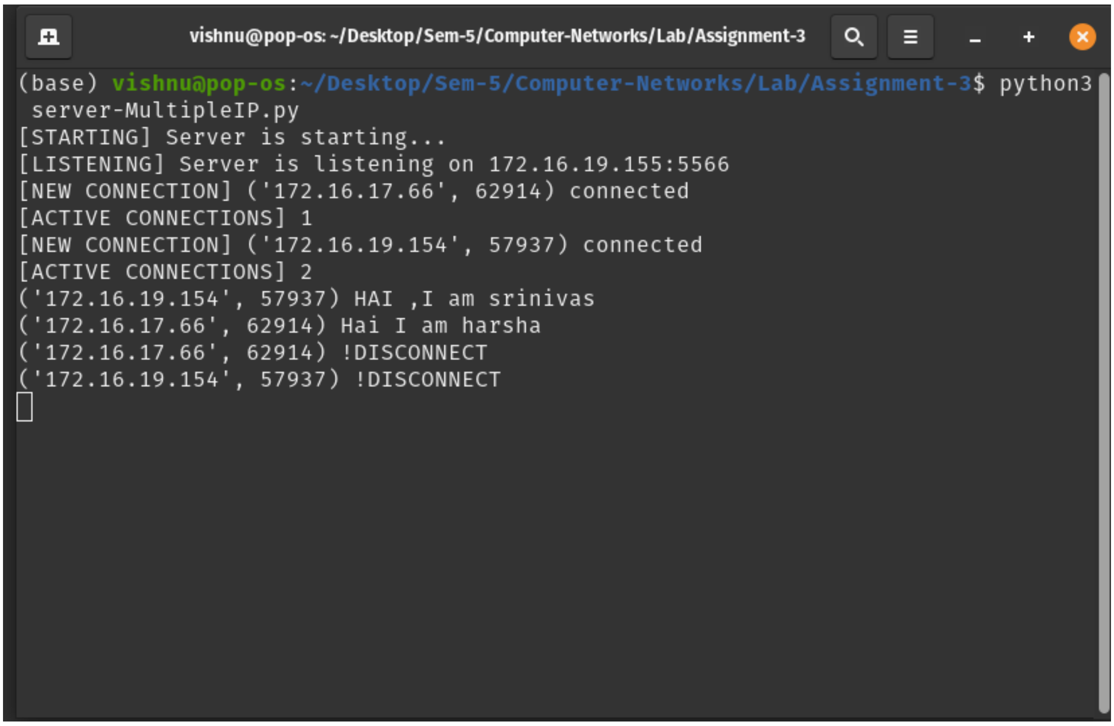

vishnu@pop-os: ~/Desktop/Sem-5/Computer-Networks...
vishnu@pop-os: ~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...
vishnu@pop-os: ~/...

```
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Ass
ignment-3$ python3 server.py
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.1.1:5566
[NEW CONNECTION] ('127.0.0.1', 44960) connected
[ACTIVE CONNECTIONS] 1
[NEW CONNECTION] ('127.0.0.1', 54938) connected
[ACTIVE CONNECTIONS] 2
[NEW CONNECTION] ('127.0.0.1', 42024) connected
[ACTIVE CONNECTIONS] 3
[NEW CONNECTION] ('127.0.0.1', 45132) connected
[ACTIVE CONNECTIONS] 4
('127.0.0.1', 44960) I am Connection-1
('127.0.0.1', 54938) I am Connection-2
('127.0.0.1', 42024) I am Connection-3
('127.0.0.1', 45132) I am Connection-4
```

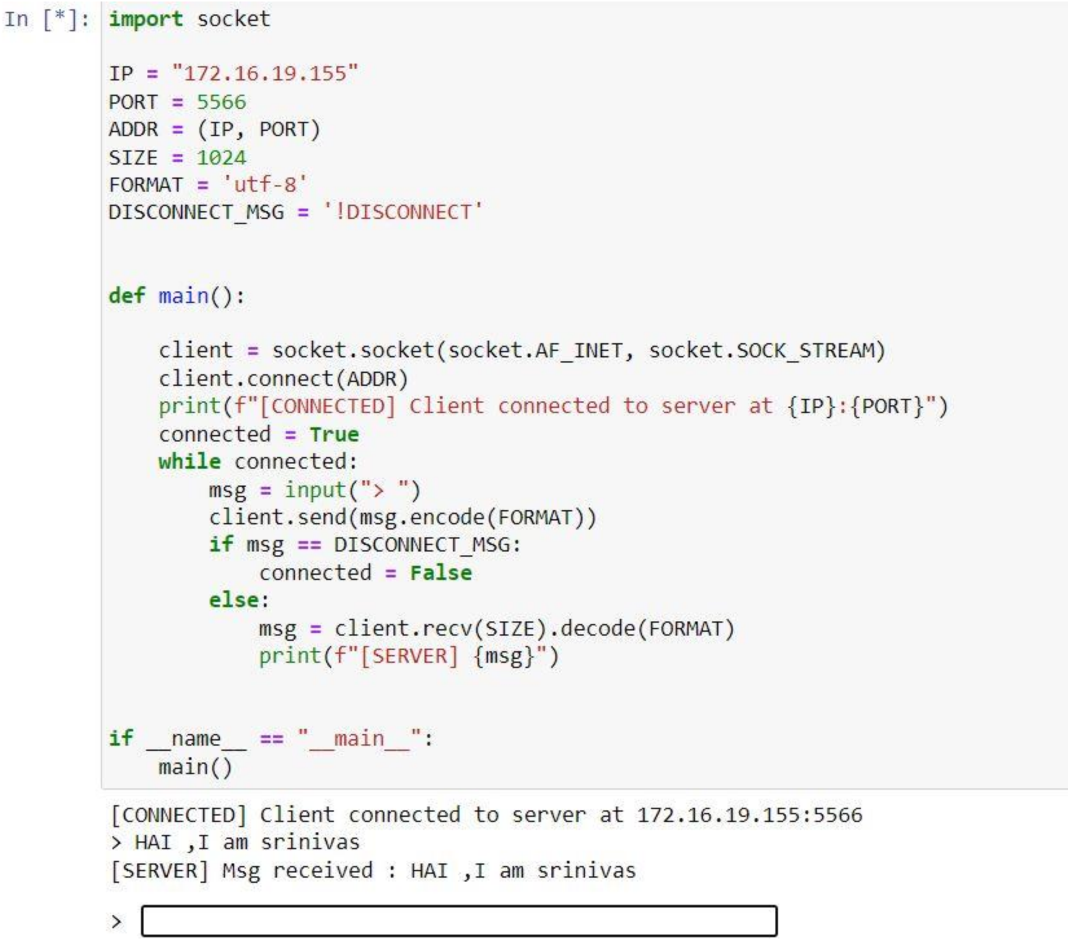
```
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3$ python3
 client.py
[CONNECTED] Client connected to server at 127.0.1.1:5566
> I am Connection-4
[SERVER] Msg received : I am Connection-4
>
```

2. Connect multiple clients with different IPs:-

Me as Server:-


vishnu@pop-os: ~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3

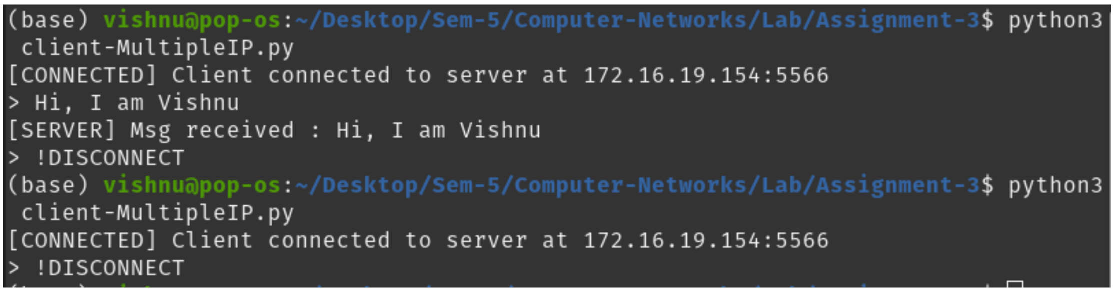
```
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3$ python3
 server-MultipleIP.py
[STARTING] Server is starting...
[LISTENING] Server is listening on 172.16.19.155:5566
[NEW CONNECTION] ('172.16.17.66', 62914) connected
[ACTIVE CONNECTIONS] 1
[NEW CONNECTION] ('172.16.19.154', 57937) connected
[ACTIVE CONNECTIONS] 2
('172.16.19.154', 57937) HAI ,I am srinivas
('172.16.17.66', 62914) Hai I am harsha
('172.16.17.66', 62914) !DISCONNECT
('172.16.19.154', 57937) !DISCONNECT
```



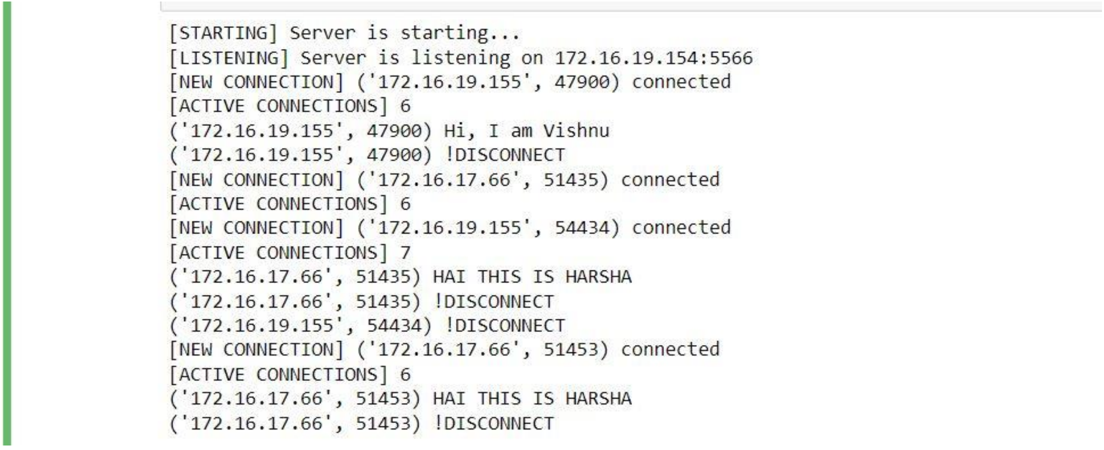
```
In [*]: import socket

        IP = "172.16.19.155"
        PORT = 5566
        ADDR = (IP, PORT)
        SIZE = 1024
        FORMAT = 'utf-8'
        DISCONNECT_MSG = '!DISCONNECT'


        def main():

            client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            client.connect(ADDR)
            print(f"[CONNECTED] Client connected to server at {IP}:{PORT}")
            connected = True
            while connected:
                msg = input("> ")
                client.send(msg.encode(FORMAT))
                if msg == DISCONNECT_MSG:
                    connected = False
                else:
                    msg = client.recv(SIZE).decode(FORMAT)
                    print(f"[SERVER] {msg}")


        if __name__ == "__main__":
            main()
```

```
[CONNECTED] Client connected to server at 172.16.19.155:5566
> HAI ,I am srinivas
[SERVER] Msg received : HAI ,I am srinivas

>
```

Me as a Client:-



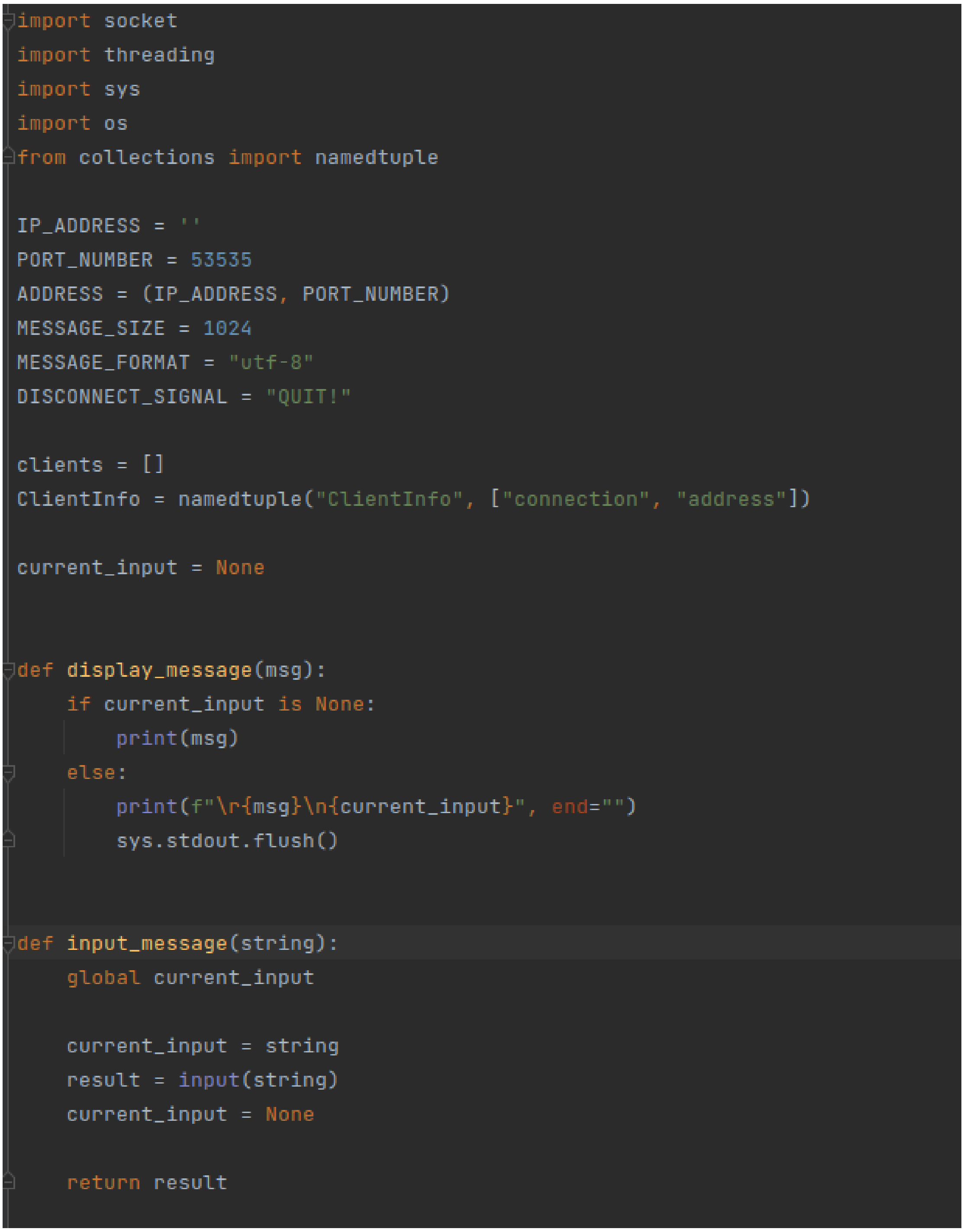
```
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3$ python3
 client-MultipleIP.py
[CONNECTED] Client connected to server at 172.16.19.154:5566
> Hi, I am Vishnu
[SERVER] Msg received : Hi, I am Vishnu
> !DISCONNECT
(base) vishnu@pop-os:~/Desktop/Sem-5/Computer-Networks/Lab/Assignment-3$ python3
 client-MultipleIP.py
[CONNECTED] Client connected to server at 172.16.19.154:5566
> !DISCONNECT
```



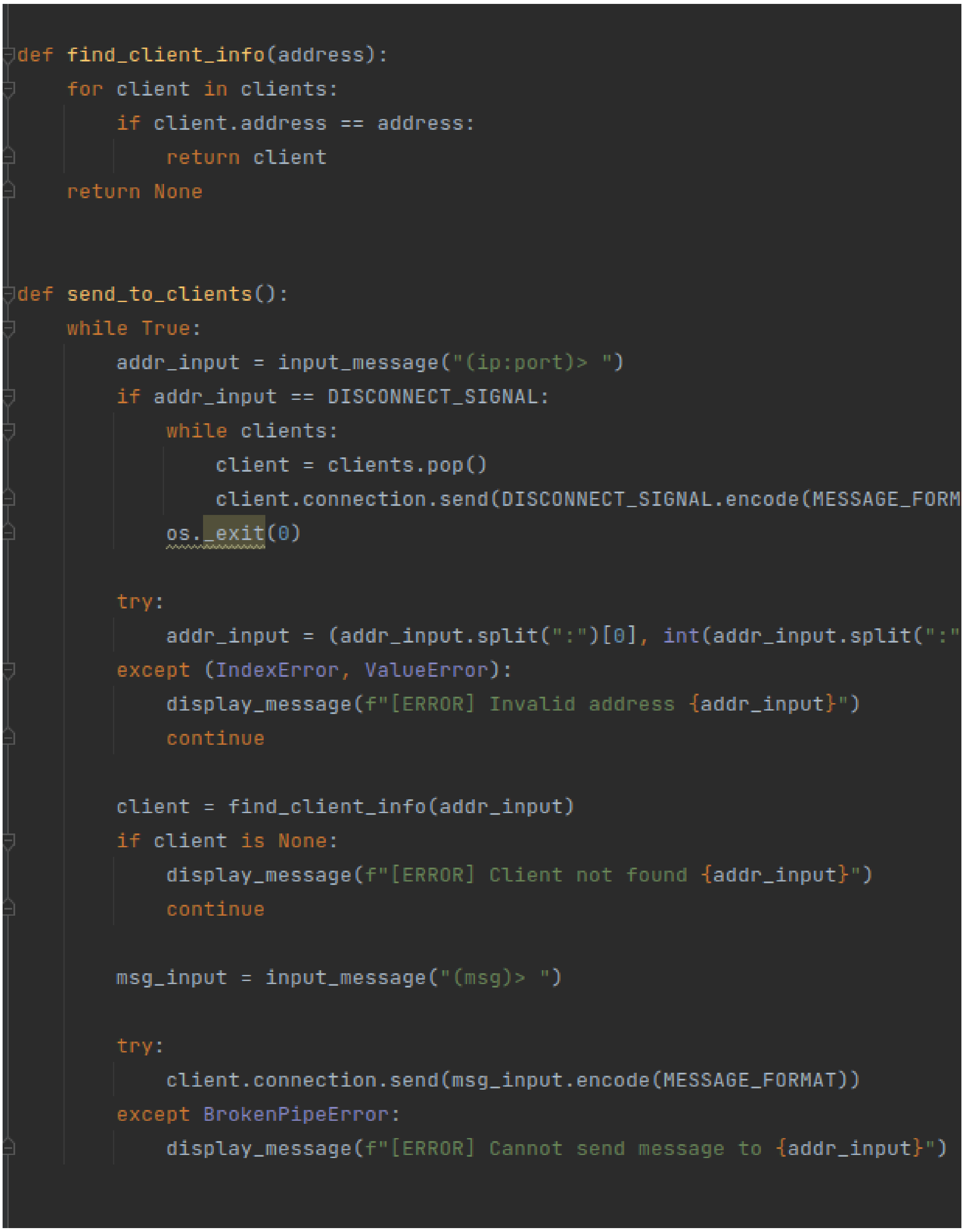
```
[STARTING] Server is starting...
[LISTENING] Server is listening on 172.16.19.154:5566
[NEW CONNECTION] ('172.16.19.155', 47900) connected
[ACTIVE CONNECTIONS] 6
('172.16.19.155', 47900) Hi, I am Vishnu
('172.16.19.155', 47900) !DISCONNECT
[NEW CONNECTION] ('172.16.17.66', 51435) connected
[ACTIVE CONNECTIONS] 6
[NEW CONNECTION] ('172.16.19.155', 54434) connected
[ACTIVE CONNECTIONS] 7
('172.16.17.66', 51435) HAI THIS IS HARSHA
('172.16.17.66', 51435) !DISCONNECT
('172.16.19.155', 54434) !DISCONNECT
[NEW CONNECTION] ('172.16.17.66', 51453) connected
[ACTIVE CONNECTIONS] 6
('172.16.17.66', 51453) HAI THIS IS HARSHA
('172.16.17.66', 51453) !DISCONNECT
```

3. Modify the program such that Server can send messages

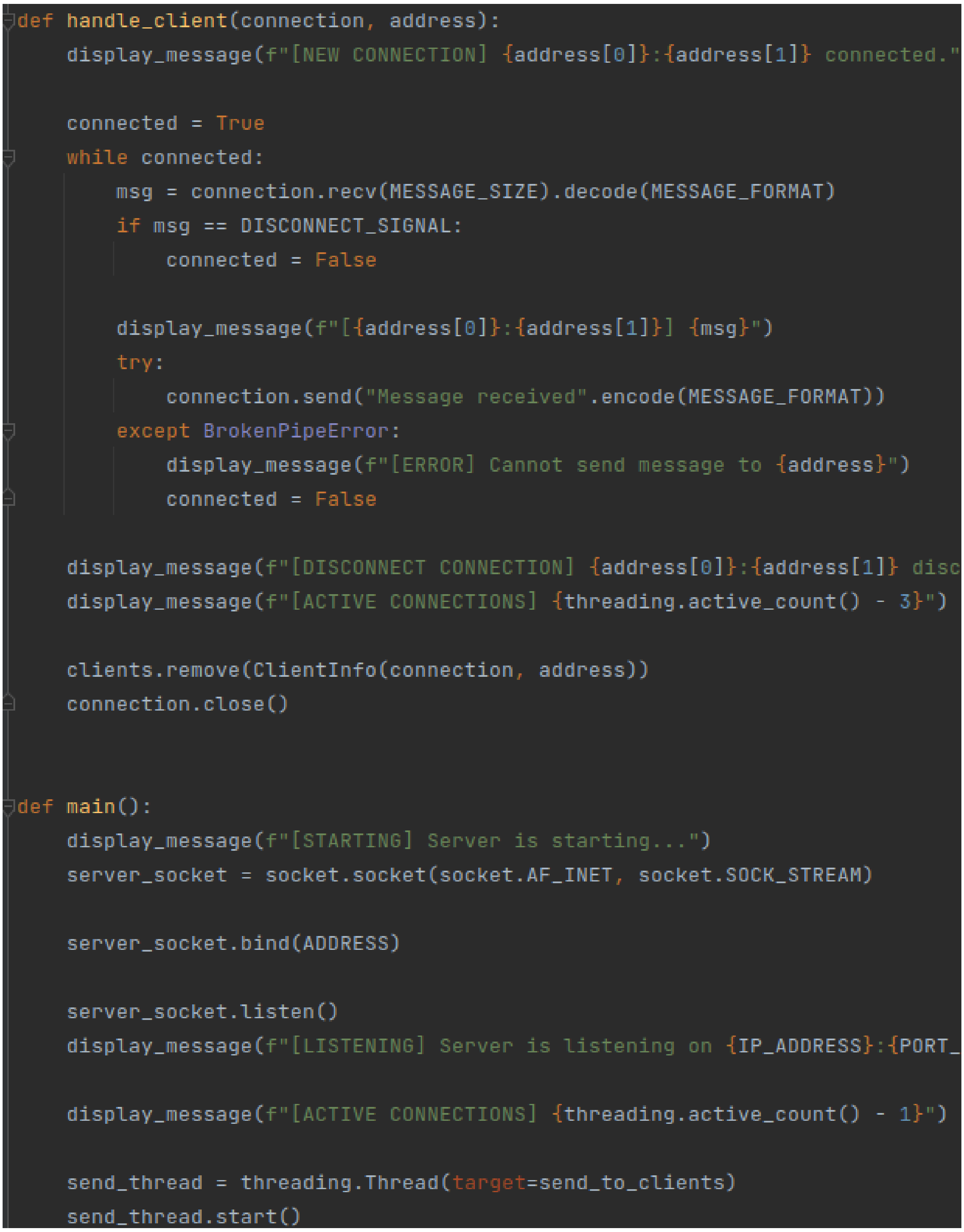
Server Code:-



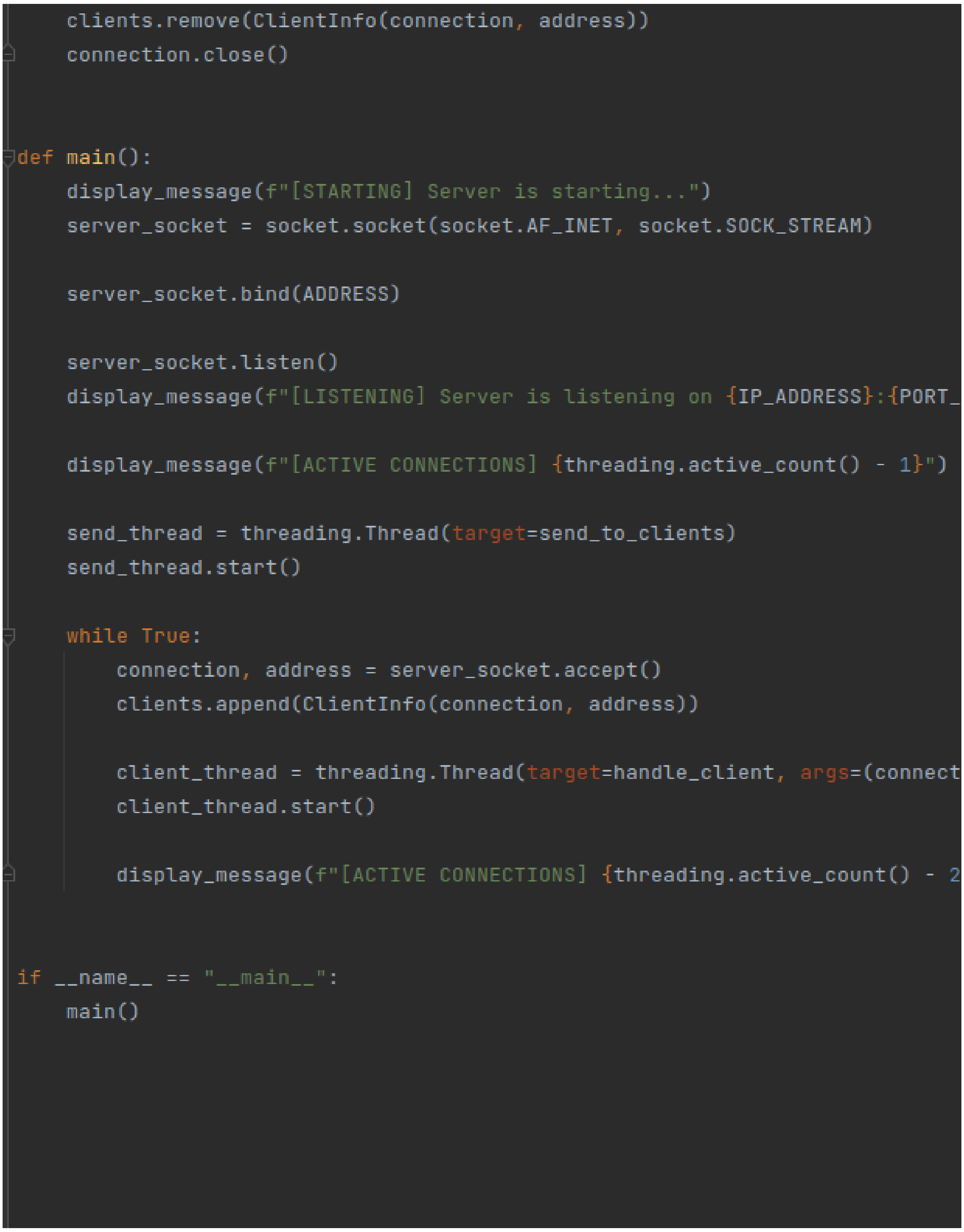
```
import socket
import threading
import sys
import os
from collections import namedtuple

IP_ADDRESS = ''
PORT_NUMBER = 53535
ADDRESS = (IP_ADDRESS, PORT_NUMBER)
MESSAGE_SIZE = 1024
MESSAGE_FORMAT = "utf-8"
DISCONNECT_SIGNAL = "QUIT!"

clients = []
ClientInfo = namedtuple("ClientInfo", ["connection", "address"])

current_input = None


def display_message(msg):
    if current_input is None:
        print(msg)
    else:
        print(f"\r{msg}\n{current_input}", end="")
        sys.stdout.flush()


def input_message(string):
    global current_input

    current_input = string
    result = input(string)
    current_input = None

    return result
```



```python
def find_client_info(address):
    for client in clients:
        if client.address == address:
            return client
    return None


def send_to_clients():
    while True:
        addr_input = input_message("(ip:port)> ")
        if addr_input == DISCONNECT_SIGNAL:
            while clients:
                client = clients.pop()
                client.connection.send(DISCONNECT_SIGNAL.encode(MESSAGE_FORM
            os._exit(0)

        try:
            addr_input = (addr_input.split(":")[0], int(addr_input.split(":"
        except (IndexError, ValueError):
            display_message(f"[ERROR] Invalid address {addr_input}")
            continue

        client = find_client_info(addr_input)
        if client is None:
            display_message(f"[ERROR] Client not found {addr_input}")
            continue

        msg_input = input_message("(msg)> ")

        try:
            client.connection.send(msg_input.encode(MESSAGE_FORMAT))
        except BrokenPipeError:
            display_message(f"[ERROR] Cannot send message to {addr_input}")
```

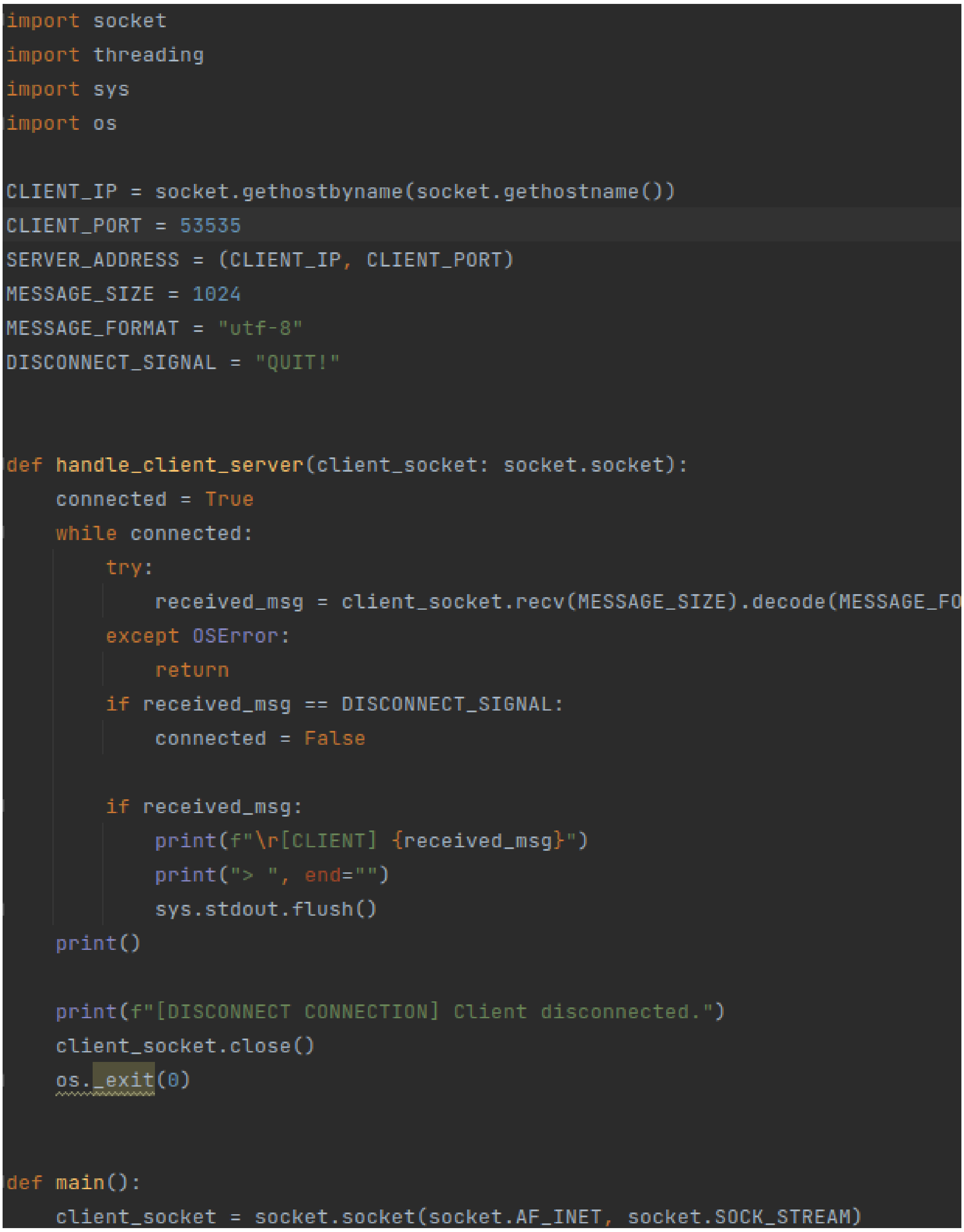


```
def handle_client(connection, address):
    display_message(f"[NEW CONNECTION] {address[0]}:{address[1]} connected."

    connected = True
    while connected:
        msg = connection.recv(MESSAGE_SIZE).decode(MESSAGE_FORMAT)
        if msg == DISCONNECT_SIGNAL:
            connected = False

        display_message(f"[{address[0]}:{address[1]}] {msg}")
        try:
            connection.send("Message received".encode(MESSAGE_FORMAT))
        except BrokenPipeError:
            display_message(f"[ERROR] Cannot send message to {address}")
            connected = False

    display_message(f"[DISCONNECT CONNECTION] {address[0]}:{address[1]} disc
    display_message(f"[ACTIVE CONNECTIONS] {threading.active_count() - 3}")

    clients.remove(ClientInfo(connection, address))
    connection.close()


def main():
    display_message(f"[STARTING] Server is starting...")
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    server_socket.bind(ADDRESS)

    server_socket.listen()
    display_message(f"[LISTENING] Server is listening on {IP_ADDRESS}:{PORT_

    display_message(f"[ACTIVE CONNECTIONS] {threading.active_count() - 1}")

    send_thread = threading.Thread(target=send_to_clients)
    send_thread.start()
```

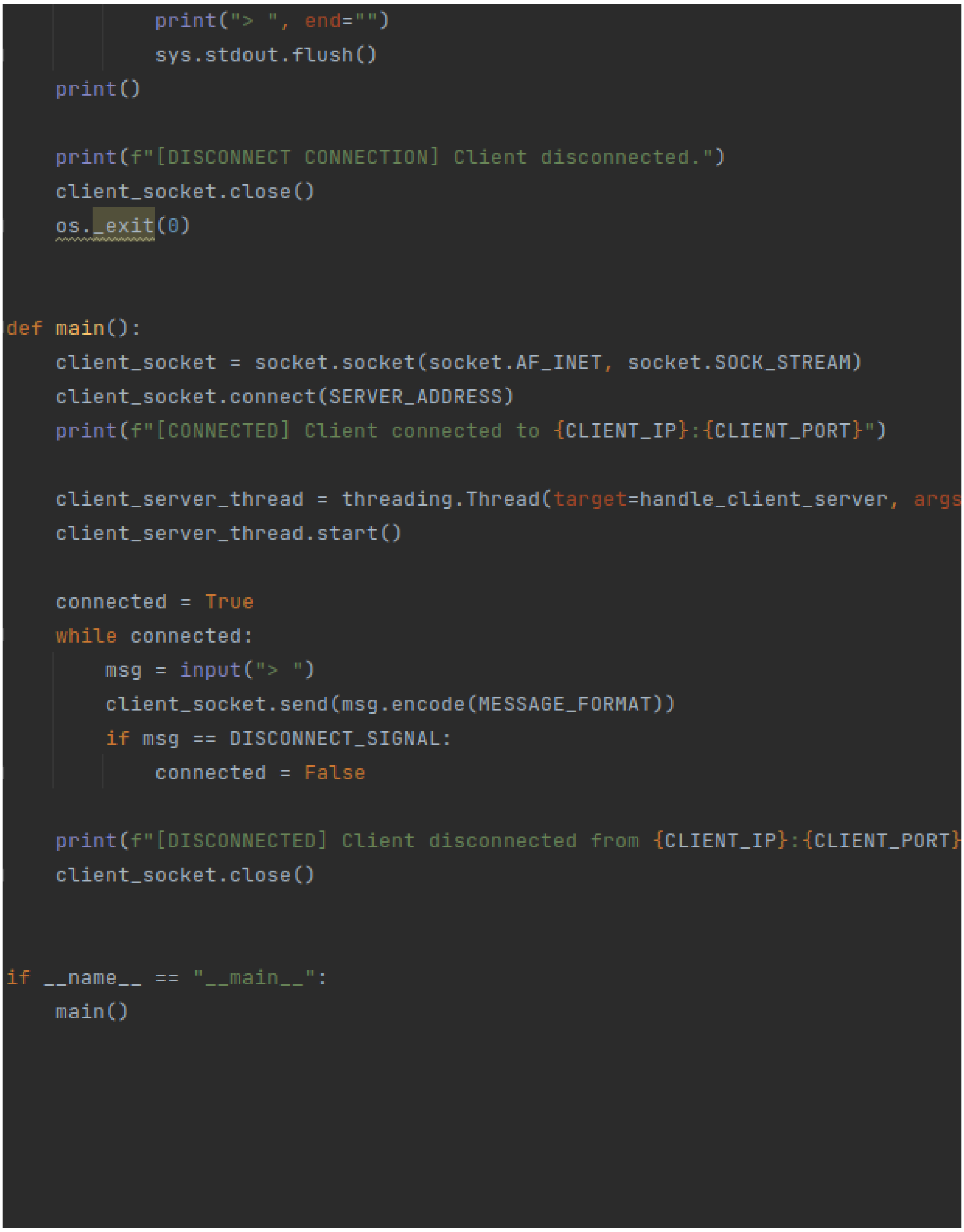


```
    clients.remove(ClientInfo(connection, address))
    connection.close()


def main():
    display_message(f"[STARTING] Server is starting...")
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    server_socket.bind(ADDRESS)

    server_socket.listen()
    display_message(f"[LISTENING] Server is listening on {IP_ADDRESS}:{PORT_

    display_message(f"[ACTIVE CONNECTIONS] {threading.active_count() - 1}")

    send_thread = threading.Thread(target=send_to_clients)
    send_thread.start()

    while True:
        connection, address = server_socket.accept()
        clients.append(ClientInfo(connection, address))

        client_thread = threading.Thread(target=handle_client, args=(connect
        client_thread.start()

        display_message(f"[ACTIVE CONNECTIONS] {threading.active_count() - 2


if __name__ == "__main__":
    main()
```

Client Code:-



```
import socket
import threading
import sys
import os

CLIENT_IP = socket.gethostbyname(socket.gethostname())
CLIENT_PORT = 53535
SERVER_ADDRESS = (CLIENT_IP, CLIENT_PORT)
MESSAGE_SIZE = 1024
MESSAGE_FORMAT = "utf-8"
DISCONNECT_SIGNAL = "QUIT!"


def handle_client_server(client_socket: socket.socket):
    connected = True
    while connected:
        try:
            received_msg = client_socket.recv(MESSAGE_SIZE).decode(MESSAGE_FO
        except OSError:
            return
        if received_msg == DISCONNECT_SIGNAL:
            connected = False

        if received_msg:
            print(f"\r[CLIENT] {received_msg}")
            print("> ", end="")
            sys.stdout.flush()
    print()

    print(f"[DISCONNECT CONNECTION] Client disconnected.")
    client_socket.close()
    os._exit(0)


def main():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```



```
            print("> ", end="")
            sys.stdout.flush()
    print()

    print(f"[DISCONNECT CONNECTION] Client disconnected.")
    client_socket.close()
    os._exit(0)


def main():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(SERVER_ADDRESS)
    print(f"[CONNECTED] Client connected to {CLIENT_IP}:{CLIENT_PORT}")

    client_server_thread = threading.Thread(target=handle_client_server, args
    client_server_thread.start()

    connected = True
    while connected:
        msg = input("> ")
        client_socket.send(msg.encode(MESSAGE_FORMAT))
        if msg == DISCONNECT_SIGNAL:
            connected = False

    print(f"[DISCONNECTED] Client disconnected from {CLIENT_IP}:{CLIENT_PORT}
    client_socket.close()


if __name__ == "__main__":
    main()
```